Qin notation generator

Candy L. K. Yiu, Jim Binkley
Department of Computer Science
Portland State University
Portland, OR USA
[candy,jrb]@cs.pdx.edu

Abstract

A Chinese character is a two-dimensional typological representation of strokes and radicals. This is unlike English letters, where there are only 26 letters which combine in one dimension. As there are thousands of Chinese characters, any Chinese character font requires more space (typically two bytes) for its computer encoding. Additionally, it takes more time for the creation of a complete font set.

Another problem is web communication. If a character is not available from the client's font set, there is no clean way to put the information in the web document. This is what originally motivated us to work on dynamic character generation.

The Qin is a Chinese musical instrument with a music specification language that serves as a good example of Chinese character stroke and radical combinations. The notation which represents the music is made with a set of characters and radicals. People who show Qin notation on the web have to scan an image and put it on the web. This project attempts to use Qin notation generation to demonstrate the possibility of Chinese character rendering.

This project is divided into two parts: the first is the Qin notation description language, and the second is the web notation generator. The description language will be based on Hanglyph, which is a syntax able to describe the hierarchical structure of Chinese characters. The Qin description language will use the basic characters and components. The rendering will construct the notation using MetaPost, based on Hanglyph. A web generator will output the pages with the Qin notation to allow communication over the internet.

1 Introduction to the Guqin

The guqin [5] (old zither) has a long history in China. It is mentioned in the Book of Odes (Shi Jing) and Confucius is said to have played it, thus we know it existed long before 200 B.C.E. Some have said that the shape of the instrument has not changed since the late Han dynasty, which is roughly two thousand years ago, between 200–500 C.E. Although many of the current pieces played on the Qin are from the Ming and Qing dynasties and may be "only" 200 to 500 years old, some pieces exist that may have started musically in B.C.E. times; for instance, the famous piece "Flowing Water" (Liu Shui). This piece is said to have been created by the legendary Qin player Bo Ya in the Spring and Autumn period. Flowing Water was included on the Voyager satellite launched in 1977, as played by the famous 20th century Qin master Guan Ping-hu. Thus the piece has been performed by Qin players for three millennia.

The Qin belonged to the old scholar class who ruled China until the fall of the Qing dynasty in 1911. The scholar class was said to practice "qin qi shu hua", "the four arts of the gentleman":

qin, the art of playing the qin
qi, the art of playing go
shu, the art of calligraphy
hua, the art of painting

They celebrated the Qin in poetry and in painting. Emperors played it, poets such as Li Bo mentioned it in poetry, and painters would often include it in a painting. The scholar class invested the Qin with an ideology that could manifest itself as a subtle form of reflective meditation through playing or view the Qin as a physical object of connoisseurship. Much of the music for the Qin is said to be thematic in the sense that it is related to nature

This work was supported as part of the Google 2005 Summer of Code, http://code.google.com/summerofcode.html.

(and Taoism), including pieces like Gao Shan (High Mountains), Liu Shui (Flowing Water), and Ping Sha Lo Yan (Geese at the Seashore). The Qin certainly has been influenced by Taoism, however other songs include thematic material taken from history, philosophy, and even occasionally romance.

As the Qin belonged to the so-called "writing class", scholars and musicians collected Qin songs, which they published in books called "qin-pu" or Qin handbooks. Hundreds of qin-pu exist with many songs (not all of which are currently played). Although there are a few individual songs on paper that predate the early Ming dynasty, in general most traditional books of songs date from around 1425 to the time of the Qing dynasty's collapse.

2 Qin tablature and the typesetting problem

The nature of traditional written Qin music is a gesture-oriented tablature. Complex symbols describe the motions of the left and right hands. Scholars believe that at some point between the Tang and Song dynasties a set of simplified Chinese characters was created as a shorthand form. These symbols took an old longhand verbose set of instructions for the left hand and right hand written in classical Chinese and combined them into a more terse form. Thus a composite Qin symbol was created that looks like an ordinary Chinese character, but is actually more variable internally than an ordinary Chinese character.

In a sense, this composite symbol is more like a sentence's worth of instructions in classical Chinese. It tells the player how to make a note with a combined left and right hand gesture. The basic components were taken from existing Chinese characters and could be combined and recombined to create simple and complex gestures that might, for example, say (see figure 1):

- using the left-hand thumb, at string 3, at stop position 10, play the string with the right-hand middle finger while pulling the string towards the player.
- follow this by sliding the left hand up to position 9 and do nothing with the right hand.
- now with the left hand not pressing any strings, with the right hand index finger play string 6 out from the player.

The "sentences" above are expressed with Qin symbols in a very concise form, requiring only a few symbols, as we can see in the figure.

The problem for typesetting is that any given symbol has a high degree of internal variability. The

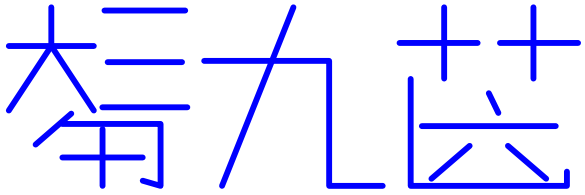

Figure 1: Example qin notation

Qin has seven strings and thirteen "stops" or left-hand position markers. For example, even though the first instruction above is relatively simple, we could change the string (1 to 7) and the left-hand stop position (1 to 13). This leads to 100 or so different symbols. We could also change the right-hand playing technique. One recent qin-pu from the 19th century [6] has about 50 separate sub-symbols for right-hand techniques and about 80 for left-hand techniques. Each stop position itself may be subdivided into 100 different parts. Thus a purportedly simple combination, without fancy glissando techniques, can easily lead to thousands of different combinations. This may be good for calligraphic artistry, but it is bad for font designers.

In short, although the traditional Qin tablature is made up out of component Chinese characters, the combined results may take many forms — and of course no existing Chinese font set would have them or be able to deal with the resulting complexity. Even today it is normal in the printing of modern Qin music for the transcriber to use traditional ink-based calligraphy to write out the Qin music and then somehow photographically insert the images onto the printed page.

From the point of view of font creation, there are many difficulties. For one, a composer could invent a new symbol. Furthermore existing handbooks in some cases document the individual character components, but the handbooks do not always agree on the atomic class of symbols! In addition, older handbooks from the Ming dynasty may use some symbols that have more or less dropped out of use in later handbooks. So although the Qin tablature system is rich in terms of tradition and semantic expression, it is fair to say it is not easily amenable to machine-based typography.

3 Structure of Qin notation

Qin notation is similar to Chinese characters. The notation can be decomposed into a number of parts called *components*. Each component consists of a number of strokes. Qin notation has a fixed number of components. Because of the fixed set of compo-

nents, we can decompose Qin symbols in three ways: top-bottom, left-right, inside-out.

Figure 2 shows example Qin notation symbols which have been decomposed into all three components. In the figure, the four areas named *A, B, C* and *D* represent four components. *A* and *B* is a left-right relation which forms one element *AB*. *C* and *D* is a inside-out relation which forms another element *CD*. Now we can see that elements *AB* and *CD* form a top-bottom relation which gives you the complete symbol structure.

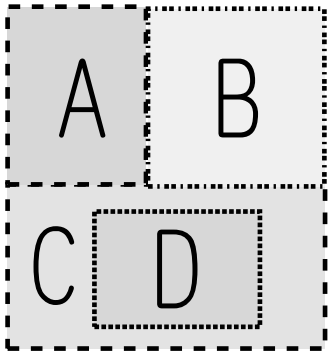

Figure 2: Example of Qin notation structure

Using the three combination operations above, we can build a hierarchical model of Qin notation based on the structure of the notation. Figure 3 shows an example which contains all 3 combinations. Figure 3 shows a Qin notation example which used the same combination structure in 2, and Figure 4 shows the hierarchical tree model after decomposing the Qin notation.

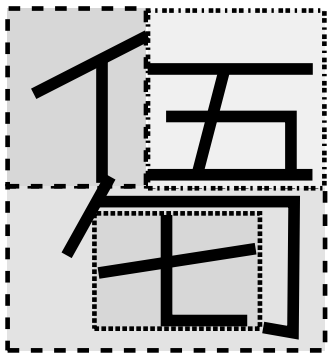

Figure 3: Qin notation decomposed elements

Thus we can use this model to build and combine single or composed elements into a new notation. We abstractly define the three combinations of the elements as follows:

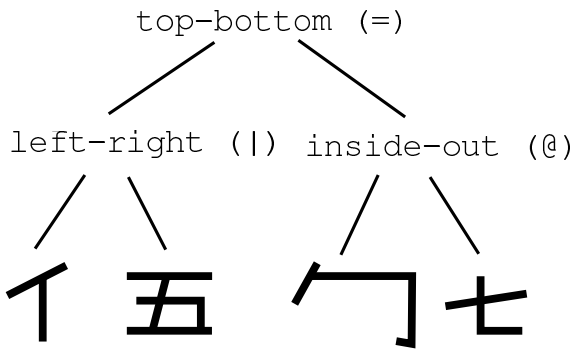

Figure 4: The hierarchical Qin notation tree

1. top-bottom: combine any two elements with one on the top of another.
2. left-right: combine any two elements with one on the left and another on the right.
3. enclosed: put one element inside another element, which has defined the enclosed area. This combination is only used when the outer element has an enclosed area.

As a result, the Qin notation generator structure is represented by a tree structure. Each internal node in the tree is any one of the above combinations (operators). The leaves of the tree are the basic pre-defined elements.

4 A Qin notation description language

In 2003, in papers entitled "Chinese Character Synthesis System using METAPOST" [3] and "Typesetting Rare Chinese Characters in LaTeX" [4], we defined a Chinese description language called Hanglyph (Chinese Description Language). This language can be used to give any Chinese character a typological representation. In Hanglyph, there are five operators and the system is based on strokes as a basic unit. In the full Hanglyph, there are also optional relations which can be used to specify more detail in the relation between the operands such as width, height, and alignment.

Our present Qin notation description language uses Hanglyph to describe the Qin notation. Qin notation, complex as it is, uses only a small subset of the more than 60,000 Chinese characters known, so it becomes a somewhat simpler problem. Therefore we only choose three operators without using any optional relations for the description of the Qin notation. However, to achieve this reduction, we require more work and information from the element library and a smarter generator engine. In the next section we will discuss the approach needed to achieve this goal.

Here is a formal definition of the Qin notation description language:

⟨*Qin Notation*⟩ ::= ⟨*expr*⟩+
⟨*expr*⟩ ::= ⟨*Qin_char*⟩ ⟨*Qin_char*⟩ ⟨*parallel_operator*⟩
| ⟨*en_Qin_char*⟩ ⟨*Qin_char*⟩ ⟨*enclosed_operator*⟩
⟨*parallel_operator*⟩ ::= ⟨*top_bottom_operator*⟩
| ⟨*left_right_operator*⟩
⟨*top_bottom_operator*⟩ ::= =
⟨*left_right_operator*⟩ ::= |
⟨*enclose_operator*⟩ ::= @
⟨*Qin_char*⟩ ::= ⟨*en_Qin_char*⟩
| ⟨*basic_Qin_char*⟩
⟨*en_Qin_char*⟩ ::= predefined element allowing enclosed
⟨*basic_Qin_char*⟩ ::= predefined basic element

5 Basic components

Traditionally there are two kinds of basic components which are taught for Qin notation in handbooks. These components are combined to make composite symbols and can be said to belong to two sets: one for the left hand, and one for the right hand. Left hand and right hand sets of symbol elements are thus predefined. Many of the elements are Chinese character components or simplified versions of them.

Our basic elements come from a late Qing dynasty Qin handbook [6]. We use a pinyin derived variable name for the elements so that they can be encoded with our basic operators. There are 87 left hand components and 47 right hand components. Figure 5 shows some examples of basic components.

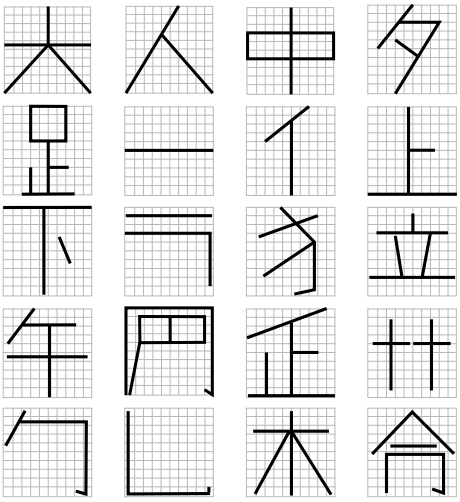

Figure 5: Examples of basic components

6 Notation generator

The notation generator generates the Qin notation in a geometric format such that we can convert the result into an image or font. It takes the Hanglyph input and generates the graphical representation as output. Figure 6 shows the structure of the Qin notation generator.

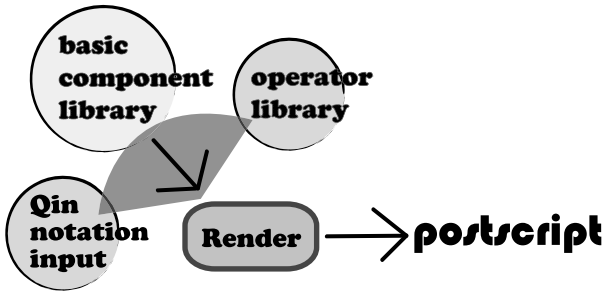

Figure 6: Qin notation generator architecture

The generator requires the basic components library and operation library, which are both written in MetaPost. We thus can divide the process into three stages:

1. MetaPost translator: Translates the Hanglyph input of the Qin notation into MetaPost output.
2. Qin notation in PostScript format: The generator uses the basic component library and performs desired operations using the operation libray to generate the Qin notation output in PostScript format.
3. Converter: The last step is to convert the PostScript format into whatever other format may be desired. In this project, we use a third-party tool, namely ImageMagick, to convert to png format in order to allow web display.

6.1 Basic component library

The basic component library is written as MetaPost macros, named for each component pinyin symbol. Figure 5 shows examples of basic components. For each component, we have control points of the path representing the component.

We also have a path creation macro to allow scaling the control points. This means when one scales the control point, one does not scale the path at the same time. Thus we achieve separation of control points and paths. This is very important because otherwise different and undesirable results may occur.

6.2 Operation library

As mentioned before, we use only three operations: top-bottom, left-right and enclosed. The operation must determine the ratio of the two components and related spacing occupied by each component. In addition, the space between the two components also has to be determined, in order to check whether two components are touching each other.

Thus, the component library contains two properties, *enclosed area* and *surrounding*, to allow the operation to obtain the needed information for each component.

6.2.1 Enclosed area property

The enclosed area specifies the possible area for an element which can contain another element. The operation uses this information to scale the inner element as needed, combining both elements to generate output. Figure 7 shows three examples of possible enclosed components. The gray area represents the location of the inner element.

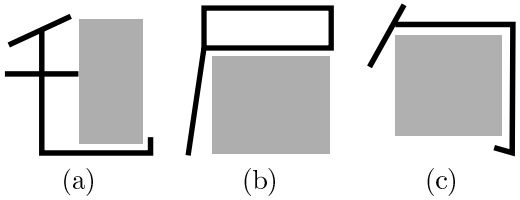

Figure 7: Examples of areas available for enclosed components

6.2.2 Surrounding property

The surrounding properties are defined in terms of four directions:

1. North (N): top side of the element.
2. East (E): left side of the element.
3. South (S): bottom side of the element.
4. West (W): right side of the element.

Furthermore, each direction can be flat or non-flat. Figure 8(a) shows the definition of each face direction. Zero represents flat and one represents non-flat. Below we show the equation to determine if spacing between two elements is needed.

$$space = 1, \text{ if } s_1 = s_2 \quad (1)$$

$$space = 0, \text{ if } s_1 \neq s_2 \quad (2)$$

For the left-right operator shown in figure 8(b), we consider the east face (right side) of the first operand and west face (left side) of the second operand. If the top-bottom operator is used as in figure 8(c), we consider the south face (bottom side) of the first operand and north face (top side) of the second operand.

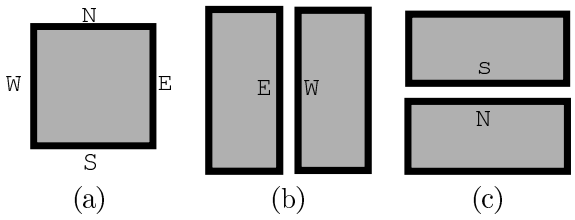

Figure 8: The direction of each component has a defined face: (a) face definitions; (b) left-right operations on faces; (c) top-bottom operations on faces

The Qin notation generator can use these properties and rules to determine the needed spacing between two operands. The equation above states that if there are two faces which have the same properties, either flat or non-flat, we insert space. Otherwise, two elements touch each other by default.

6.2.3 Area ratio

Our next concern is to determine the ratio of the visual area of the two operands. We use the elements below to consider how to estimate the ratio:

1. The number of strokes in each component.
2. The total length of the strokes in each component.
3. The width of each component.

The first row of figure 9 (1a,1b,1c) shows different possible area ratios occupied by each operand. The Qin notation generator has to calculate and determine the ratio of each operand in order to give a good visual output for the notation. The second

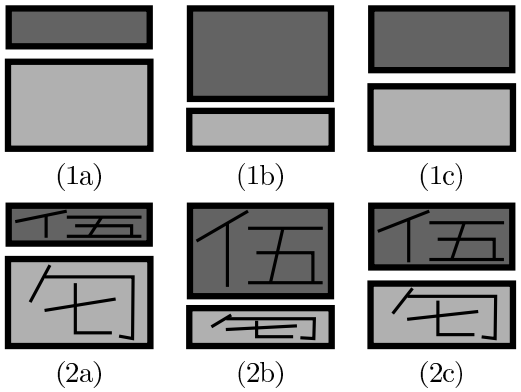

Figure 9: Spacing operands: first row shows generic ratios; second row shows possible characters; with (2c) being the best result.

row of figure 9 (2a,2b,2c) shows example character for different ratios. In this example, (2c) is the best output. In our experiment, the above rules to calculate the notation work well.

7 Future work

We can further fine-tune the Qin notation generator by studying previously printed qin-pu and the symbols used in them to do a better job estimating a more aesthetic ratio between our operands. After we generate the notation, the layout of the music piece is another interesting topic as well. To typeset an entire piece of music, we need a language for describing the layout of the piece. This could, for example, combine Western staff notation with the traditional Qin notation in order to provide necessary information about the tempo of a piece. We hope to continue this project in the future [1].

References

[1] Candy L.K. Yiu, Portland State University, Qin notation generator. `http://web.pdx.edu/~candy/qin`.

[2] Candy L.K. Yiu, Wai Wong. *Chinese Character Synthesis*. ACM PG Conference, Hong Kong, China. January 25, 2003.

[3] Candy L.K. Yiu, Wai Wong. *Chinese character synthesis using METAPOST*. In proceedings of TUG 2003, *TUGboat* **24**(1), 85–93 (2003). `http://tug.org/TUGboat/Articles/tb24-1/yiu.pdf`.

[4] Wai Wong, Candy L.K. Yiu, and Kelvin C.F. Ng. *Typesetting Rare Chinese Characters in LaTeX*. In proceedings of the 14th European TeX Conference (EuroTeX 2003), *TUGboat* **24**(3), 582–587 (2003). `http://tug.org/TUGboat/Articles/tb24-3/wong.pdf`.

[5] `http://en.wikipedia.org/wiki/Guqin`. September 2005.

[6] `http://www.cs.pdx.edu/~jrb/chin`. Yuguzhai Qinpu, 1855, Fujian, China. September 2005.

[7] John Hobby and Gu Guoan. *A Chinese metafont*. *TUGboat* **5**(2), pp. 119–136, 1984.

[8] John D. Hobby. *A user's manual for MetaPost*. AT&T Bell Laboratories Computer Science Technical Report, No. 162, 1992. `http://cm.bell-labs.com/who/hobby/pubs.html`.

[9] Donald E. Knuth. *The METAFONTbook*. Addison-Wesley, 1986.